Software Maintenance

By
Prof. A. K Dudyala
NIT Patna

Contents

- Definiton
- Types of Software Maintenance
- Problems associated with software maintenance
- Software reverse engineering
- Factors on which software maintenance activities depend
- Software maintenance process models

Software Maintenance

- The software usage has become more due to the drastic decrease in hardware prices and portability across different platforms, it mandates the maintenance of it.

- The demand of the user community to see the existing software products run on newer platforms, run in newer environments, and/or with enhanced features has Necessitated the need for software maintenance.

Types of Software Maintenance

- **Corrective:** Corrective maintenance of a software product is necessary to rectify the bugs observed while the system is in use.
- **Adaptive:** A software product might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to interface with new hardware or software.
- **Perfective:** A software product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to customer demands, or to enhance the performance of the system.

Problems associated with software maintenance

- Software maintenance work typically is much more expensive than what it should be and takes more time than required.

- Software maintenance has a very poor image in industry. Therefore, an organization often cannot employ bright engineers to carry out maintenance work.

- The majority of software products needing maintenance are legacy products.

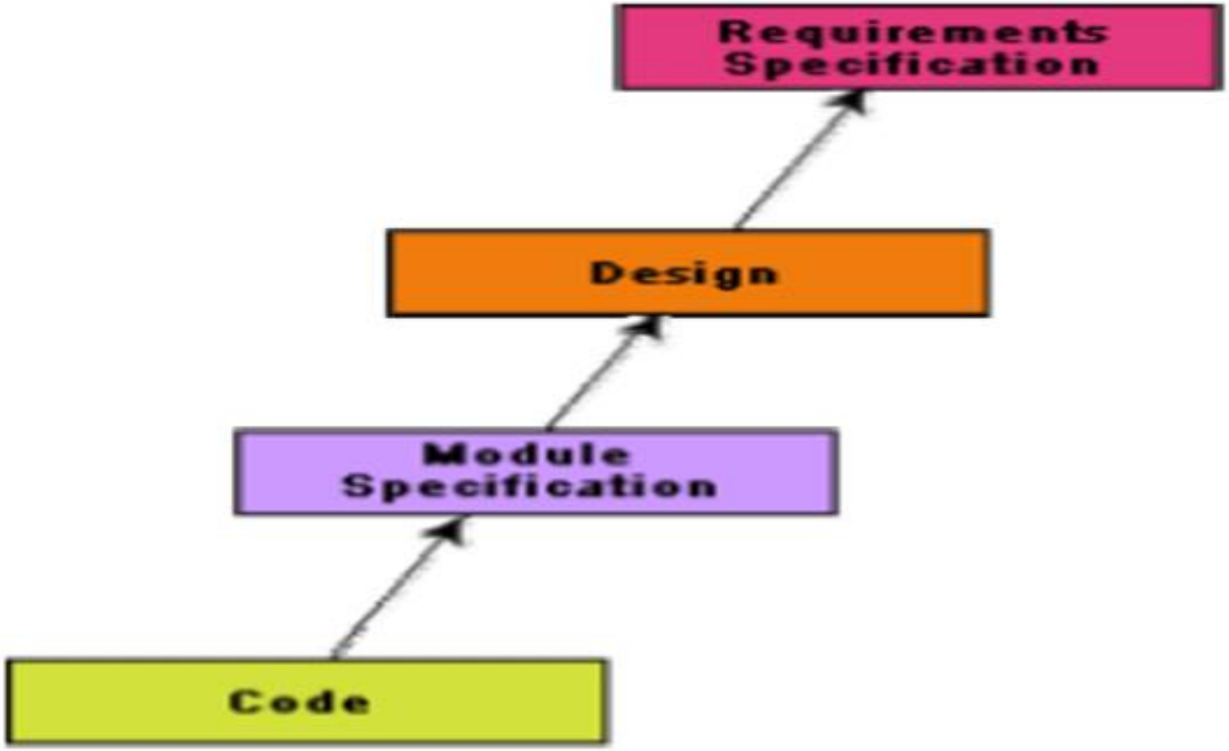
Software reverse engineering

- The process of recovering the design and the requirements specification of a product from an analysis of its code.

- The purpose of reverse engineering is to facilitate maintenance work by improving the understandability of a system and to produce the necessary documents for a legacy system.

- The first stage of reverse engineering usually focuses on carrying out cosmetic changes to the code to improve its readability, structure and understandability, without changing of its functionalities.

Reverse Engineering



A process model for reverse engineering

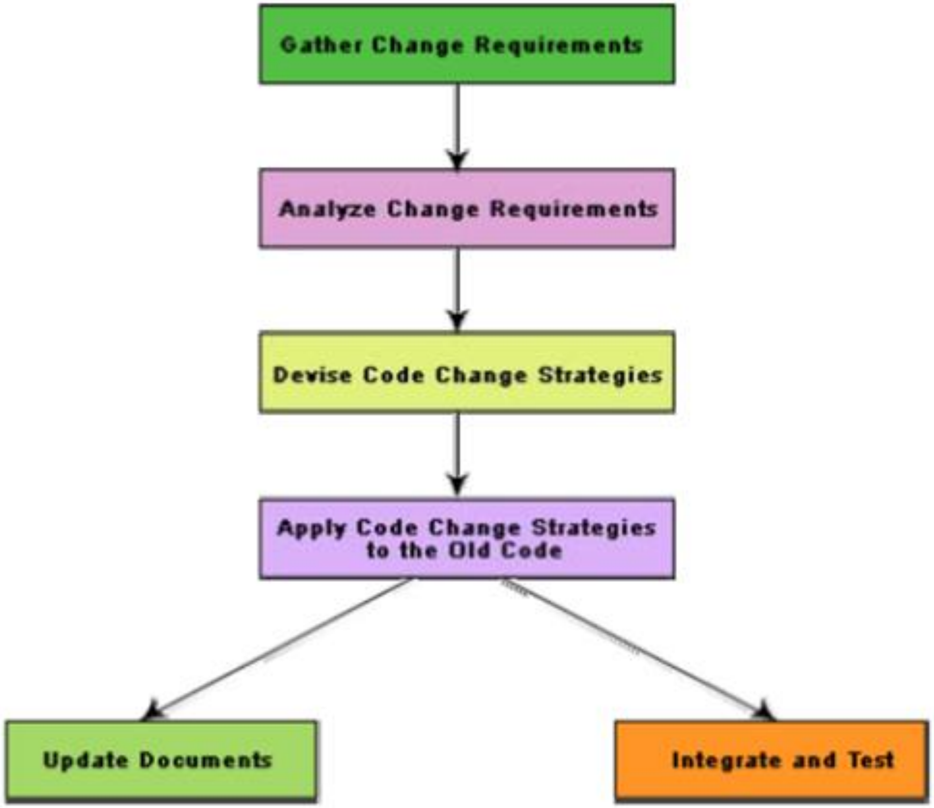
Factors on which software maintenance activities depend

- The extent of modification to the product required
- The resources available to the maintenance team
- The conditions of the existing product (e.g., how structured it is, how well documented it is, etc.)
- The expected project risks, etc.

Software maintenance process models

- Two types of models are there,
  - The first model is preferred for projects involving small reworks where the code is changed directly and the changes are reflected in the relevant documents later.
  - The second process model for software maintenance is preferred for projects where the amount of rework required is significant.
    - This approach can be represented by a reverse engineering cycle followed by a forward engineering cycle.

Software maintenance process models



Gather Change Requirements

Analyze Change Requirements

Devise Code Change Strategies

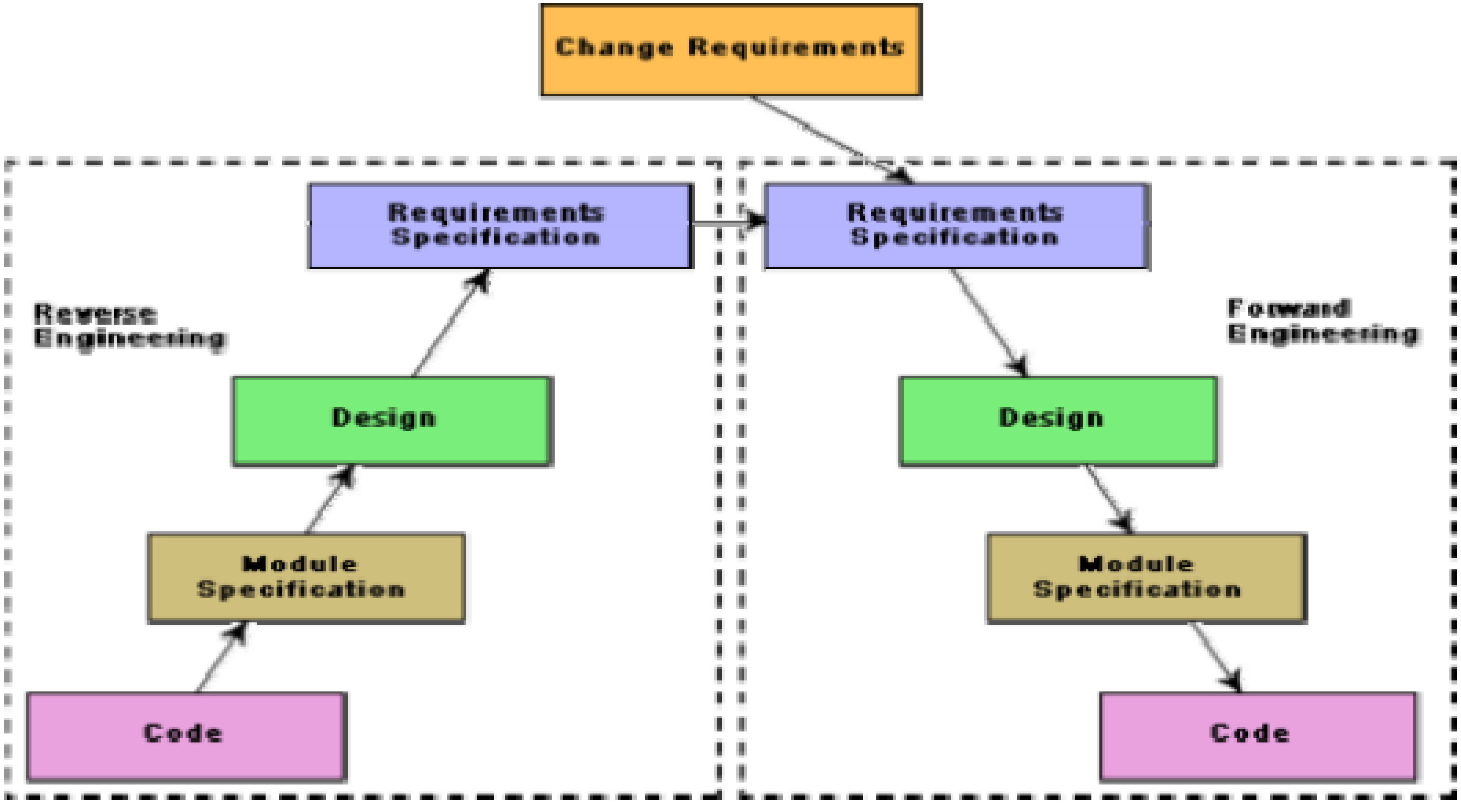
Apply Code Change Strategies to the Old Code

Update Documents

Integrate and Test

Maintenance process model 1

Software maintenance process models

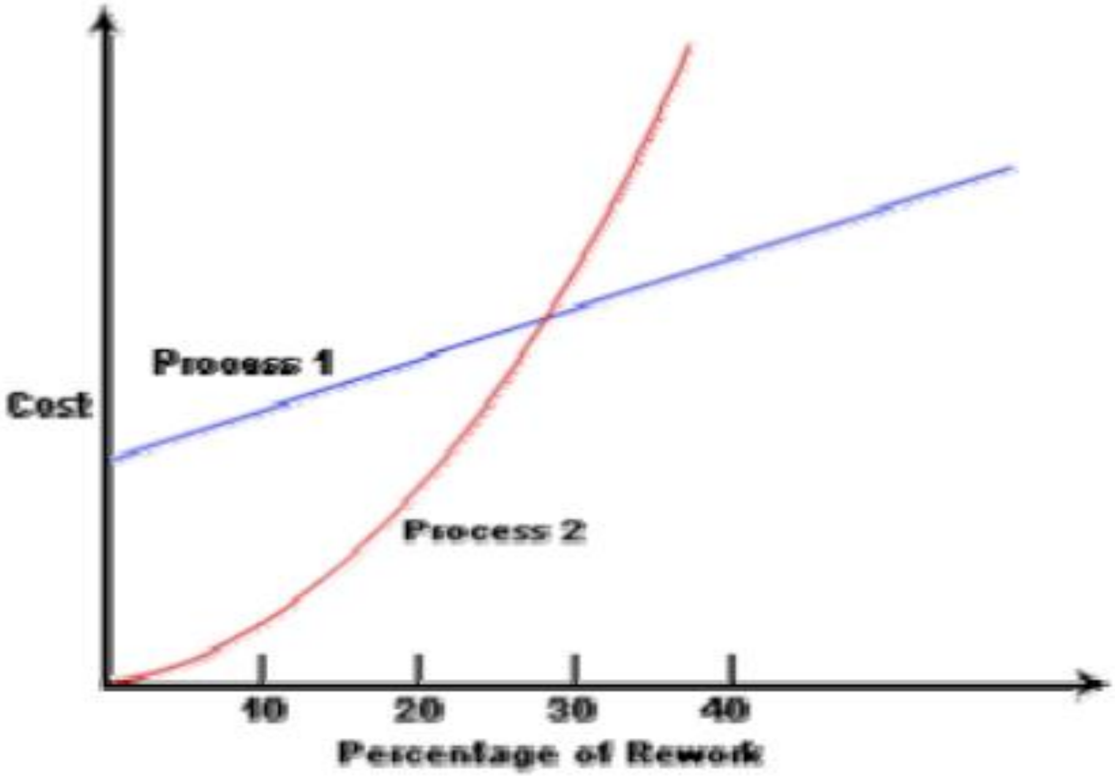


Maintenance process model 2

- An empirical study indicates that process 1 is preferable when the amount of rework is no more than 15%.

- Reengineering might be preferable for products which exhibit a high failure rate.

- Reengineering might also be preferable for legacy products having poor design and code structure.



Empirical estimation of maintenance cost versus percentage rework

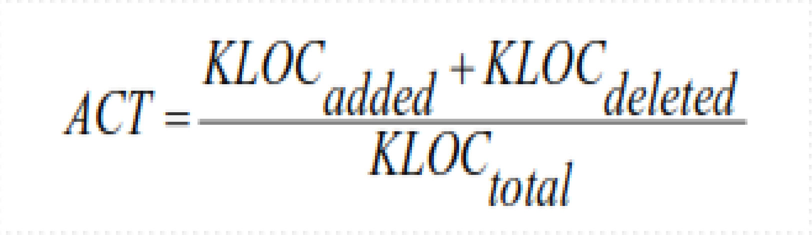
Software reengineering

- Software reengineering is a combination of two consecutive processes i.e. software reverse engineering and software forward engineering.

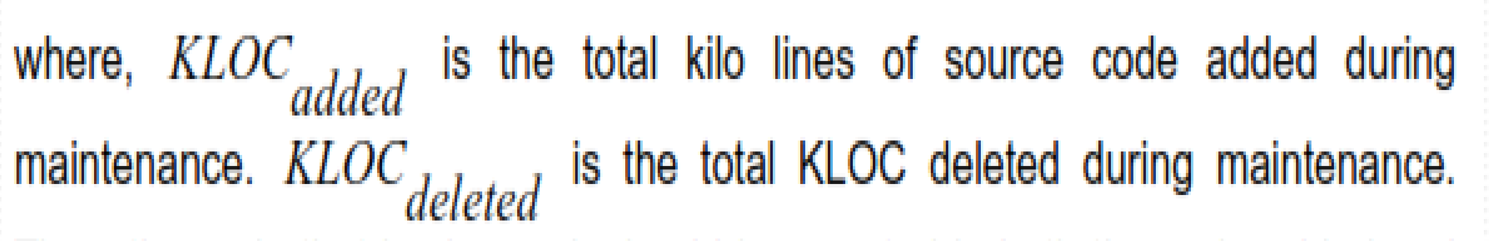
Estimation of approximate maintenance cost

- Boehm's maintenance cost estimation is made in terms of a quantity called the Annual Change Traffic (ACT).
- Boehm defined ACT as the fraction of a software product's source instructions which undergo change during a typical year either through addition or deletion.

Estimation of approximate maintenance cost



$$ACT = \frac{KLOC_{added} + KLOC_{deleted}}{KLOC_{total}}$$



where, $KLOC_{added}$ is the total kilo lines of source code added during maintenance. $KLOC_{deleted}$ is the total KLOC deleted during maintenance.

- The annual change traffic (ACT) is multiplied with the total development cost to arrive at the maintenance cost:
  - $\text{maintenance cost} = ACT \times \mathit{development\ cost}.$

Thank You